



武汉东湖学院
Wuhan Donghu College

实 验 报 告

学院：计算机科学技术专业：计算机应用技术 2025 年 4 月 14 日

姓 名	李畅	学 号	2024040731192	
班 级	一班	指导老师	孔德华老师	
课程名称	c 语言程序设计实践		成 绩	
实验名称	函数-1			

1. 实验目的

1. 理解使用函数的意义；
2. 掌握函数的定义方法；
3. 掌握函数调用中的参数传递规则。

2. 实验内容

- 1、编写一个程序，通过函数调用的方式计算 $y=|x|$ 。
- 2、在主函数中输入一个整数，在子函数中判定该数据的奇偶性后，将判定结果在主函数中输出。
- 3、编写一个函数，计算任一输入的整数的各位数字之和。其中，主函数包括输入、输出和调用该函数。
- 4、在主函数中输入两个实数，在该程序中用四个子函数实现求这两个数据的和、差、积、商，并在主函数中将计算结果输出。
- 5、用函数实现该功能：判断某年是否为闰年，输入输出信息均在主函数中实现。
- 6、编写一个函数，判断某一个四位数是不是玫瑰花数（所谓玫瑰花数即该四位数各位数字的四次方和恰好等于该数本身，如： $1634=1^4+6^4+3^4+4^4$ ）。在主函数中从键盘任意输入一个四位数，调用该函数，判断该数是否为玫瑰花数，若是则输出“yes”，否则输出“no”。
- 7、用递归法计算菲波那契数列（Fibonacci 序列）的前 20 项。
$$\begin{aligned} f(1) &= 1 & (n=1) \\ f(2) &= 1 & (n=2) \\ f(n) &= f(n-1) + f(n-2) & (n \geq 3) \end{aligned}$$
- 8、利用函数调用求两整数的最大公约数和最小公倍数。

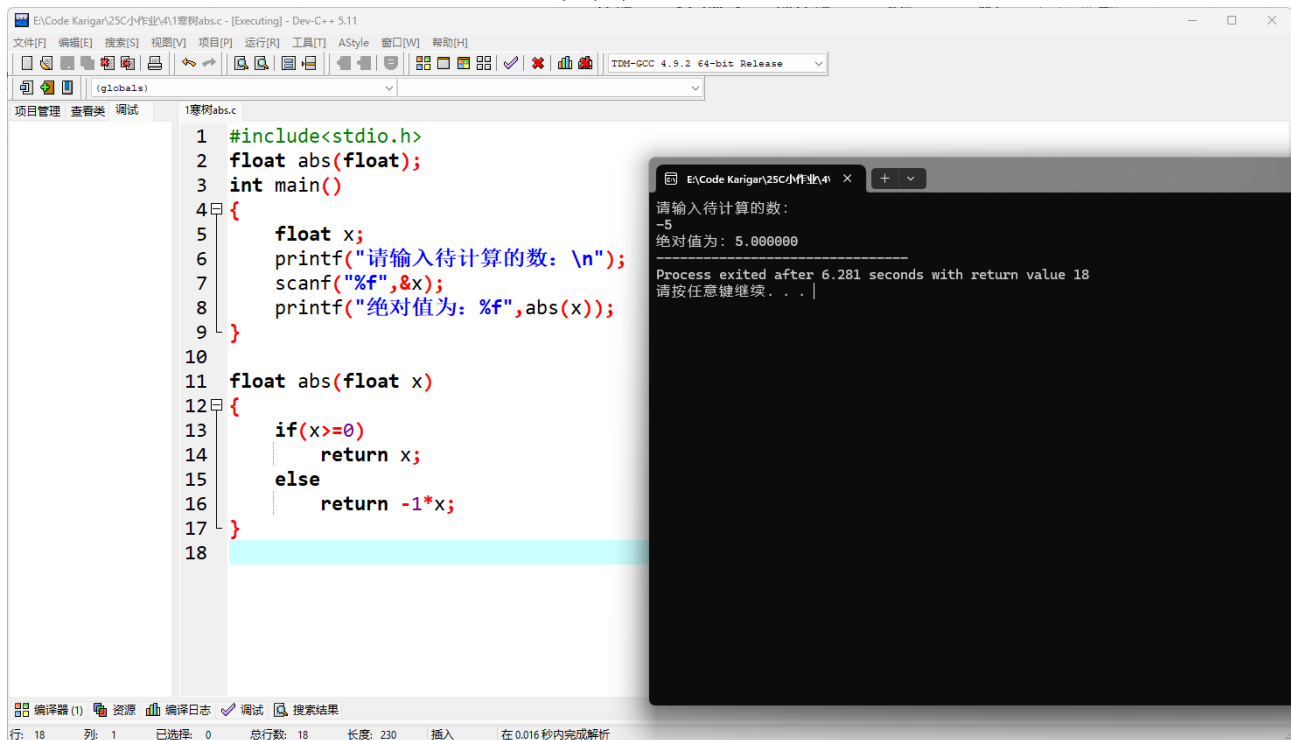
3. 实验环境

① windows 10

② Vc++6.0

4. 实验方法和步骤（含设计）

1、编写一个程序，通过函数调用的方式计算 $y=|x|$ 。



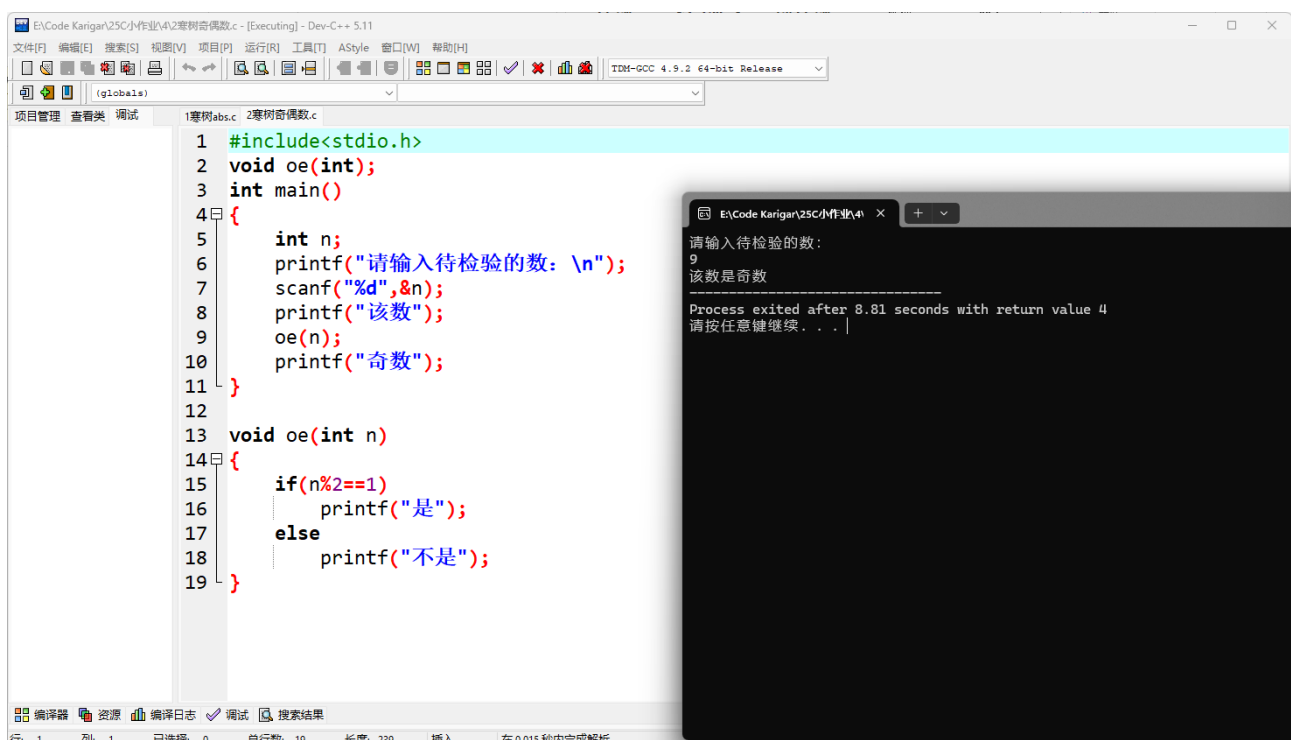
The screenshot shows the Dev-C++ IDE with a project named "1 案例abs.c". The code defines a function `abs` that calculates the absolute value of a float `x`. The `main` function prompts the user to enter a number, reads it, and prints the absolute value. The output window shows the user entering `-5` and the program outputting `绝对值为: 5.000000`.

```
1 #include<stdio.h>
2 float abs(float);
3 int main()
4 {
5     float x;
6     printf("请输入待计算的数: \n");
7     scanf("%f",&x);
8     printf("绝对值为: %f",abs(x));
9 }
10
11 float abs(float x)
12 {
13     if(x>=0)
14         return x;
15     else
16         return -1*x;
17 }
18
```

Output window content:

```
请输入待计算的数:
-5
绝对值为: 5.000000
Process exited after 6.281 seconds with return value 18
请按任意键继续. . .
```

2、在主函数中输入一个整数，在子函数中判定该数据的奇偶性后，将判定结果在主函数中输出。



The screenshot shows the Dev-C++ IDE with a project named "1 案例abs.c 2 案例奇偶数.c". The code defines a function `oe` that checks if an integer `n` is odd or even. The `main` function prompts the user to enter a number, reads it, and prints the result. The output window shows the user entering `9` and the program outputting `该数是奇数`.

```
1 #include<stdio.h>
2 void oe(int);
3 int main()
4 {
5     int n;
6     printf("请输入待检验的数: \n");
7     scanf("%d",&n);
8     printf("该数");
9     oe(n);
10    printf("奇数");
11 }
12
13 void oe(int n)
14 {
15     if(n%2==1)
16         printf("是");
17     else
18         printf("不是");
19 }
```

Output window content:

```
请输入待检验的数:
9
该数是奇数
Process exited after 8.81 seconds with return value 4
请按任意键继续. . .
```

3、编写一个函数，计算任一输入的整数的各位数字之和。其中，主函数包括输入、输出和调用该函数。

The screenshot shows the Dev-C++ IDE with a C program. The code defines a function `cardinality_sum` that calculates the sum of digits of an integer. The `main` function prompts the user for an integer, reads it, and calls `cardinality_sum` to calculate the sum of its digits.

```
1 #include<stdio.h>
2 int cardinality_sum(int);
3 int main()
4 {
5     int n;
6     printf("请输入待计算的数: \n");
7     scanf("%d",&n);
8     printf("该数各位数字之和为: %d",cardinality_sum(n));
9 }
10
11 int cardinality_sum(int n)
12 {
13     int sum=0;
14     for(;n!=0;)
15     {
16         sum+=n%10;
17         n/=10;
18     }
19     return sum;
20 }
```

The terminal window shows the execution results:

```
请输入待计算的数:
16339
该数各位数字之和为: 22
-----
Process exited after 15 seconds with return value 22
请按任意键继续. . .
```

4、在主函数中输入两个实数，在该程序中用四个子函数实现求这两个数据的和、差、积、商，并在主函数中将计算结果输出

The screenshot shows the Dev-C++ IDE with a C program. The code defines four functions: `ASMD` (Addition), `ASMD` (Subtraction), `ASMD` (Multiplication), and `ASMD` (Division). The `main` function prompts the user for two real numbers and an operator, reads them, and calls the appropriate function to calculate the result.

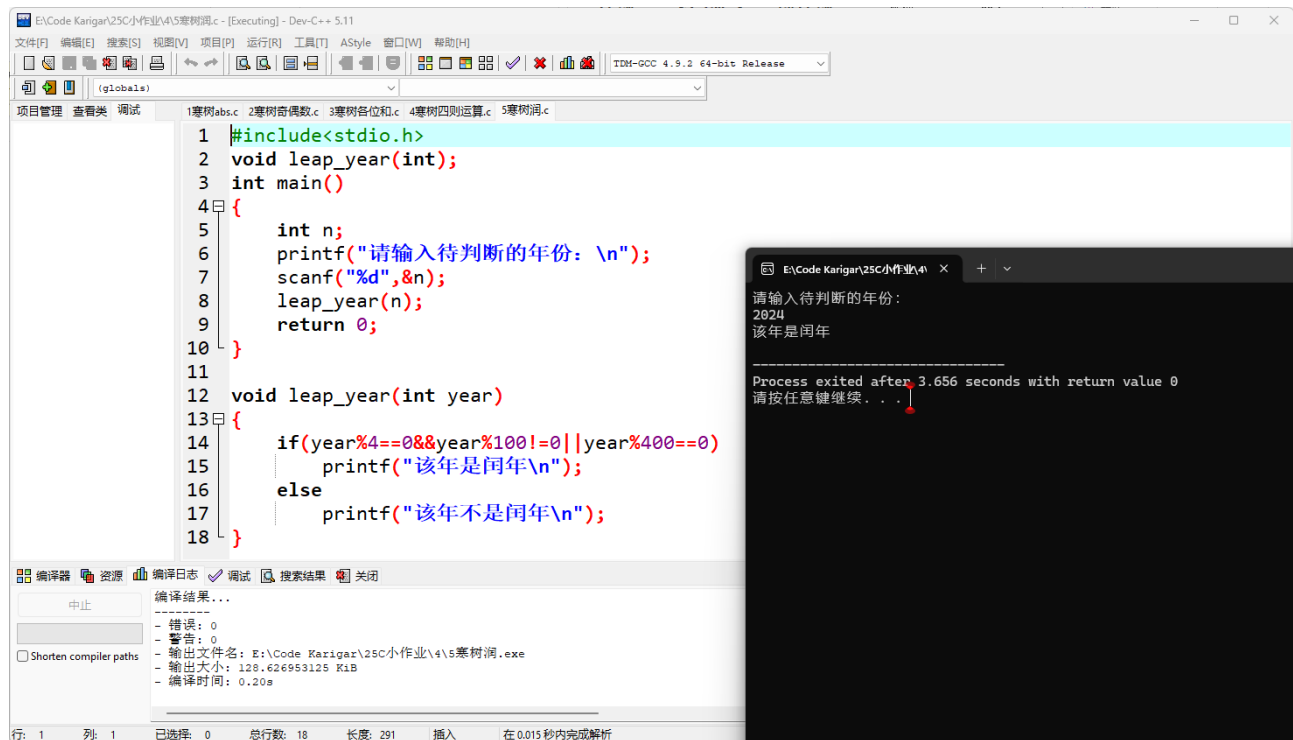
```
1 #include<stdio.h>
2 float ASMD(float,char,float);
3 int main()
4 {
5     float x,y;
6     char c;
7     printf("请按格式输入待计算的数: x ? y \n");
8     scanf("%f%c%f",&x,&c,&y);
9     printf("结果为: %.2f%c%.2f=%.2f",x,c,y,ASMD(x,c,y));
10 }
11
12 float ASMD(float x,char c,float y)
13 {
14     float f=0;
15     if(c=='+')
16         return x+y;
17     else if(c=='-')
18         return x-y;
19     else if(c=='*')
20         return x*y;
21     else if(c=='/'&&y!=0)
22         return x/y;
23     else
24     {
25         printf("输入错误请重新输入 \n");
26         return 0;
27     }
28 }
```

The terminal window shows the execution results:

```
请按格式输入待计算的数: x ? y
9/6
结果为: 9.00/6.00=1.50
-----
Process exited after 11.49 seconds with return value 22
请按任意键继续. . .
```

```
请按格式输入待计算的数: x ? y
9/0
输入错误请重新输入
结果为: 9.00/0.00=0.00
```

5、用函数实现该功能：判断某年是否为闰年，输入输出信息均在主函数中实现。

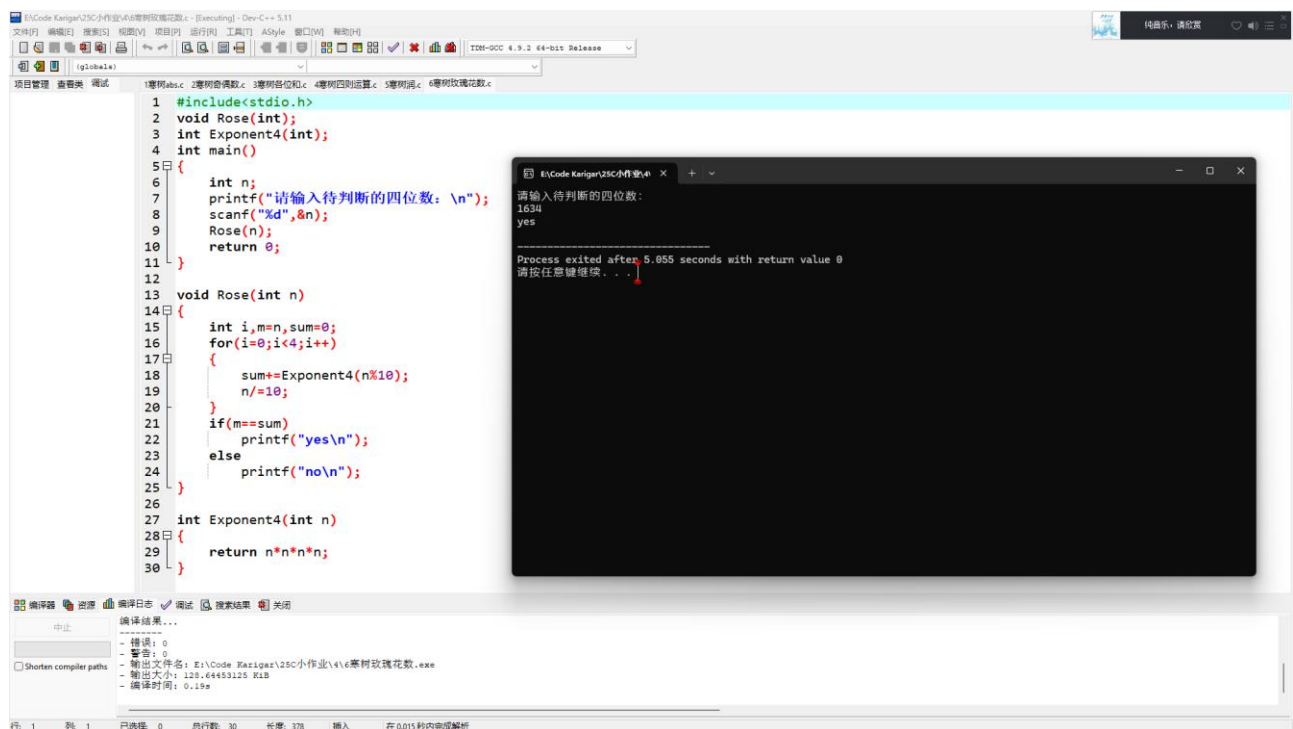


The screenshot shows the Visual Studio Code editor with a C program for leap year calculation. The code is as follows:

```
1 #include<stdio.h>
2 void leap_year(int);
3 int main()
4 {
5     int n;
6     printf("请输入待判断的年份: \n");
7     scanf("%d",&n);
8     leap_year(n);
9     return 0;
10 }
11
12 void leap_year(int year)
13 {
14     if(year%4==0&&year%100!=0||year%400==0)
15         printf("该年是闰年\n");
16     else
17         printf("该年不是闰年\n");
18 }
```

The execution output shows the program running successfully, asking for a year (2024) and outputting "该年是闰年" (It is a leap year).

6、编写一个函数，判断某一个四位数是不是玫瑰花数

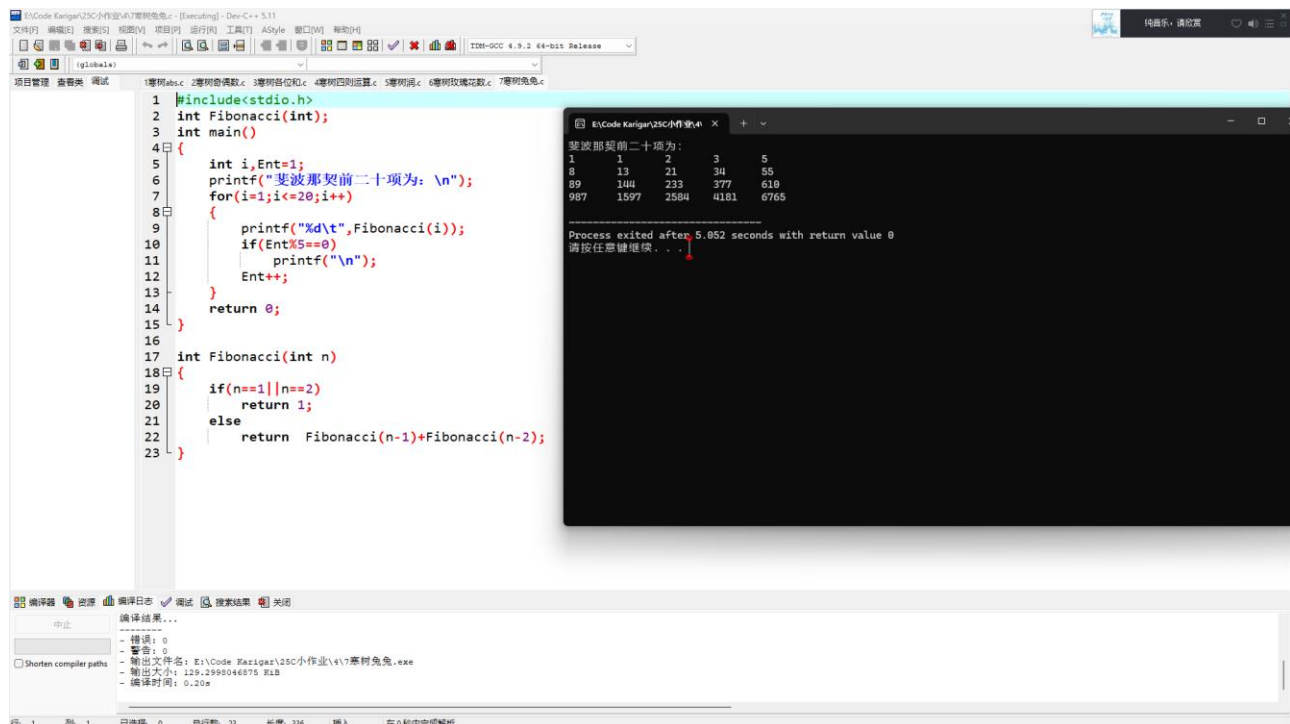


The screenshot shows the Visual Studio Code editor with a C program for Rose number calculation. The code is as follows:

```
1 #include<stdio.h>
2 void Rose(int);
3 int Exponent4(int);
4 int main()
5 {
6     int n;
7     printf("请输入待判断的四位数: \n");
8     scanf("%d",&n);
9     Rose(n);
10    return 0;
11 }
12
13 void Rose(int n)
14 {
15     int i,m=n,sum=0;
16     for(i=0;i<4;i++)
17     {
18         sum+=Exponent4(n%10);
19         n/=10;
20     }
21     if(m==sum)
22         printf("yes\n");
23     else
24         printf("no\n");
25 }
26
27 int Exponent4(int n)
28 {
29     return n*n*n*n;
30 }
```

The execution output shows the program running successfully, asking for a four-digit number (1634) and outputting "yes" (It is a Rose number).

7、用递归法计算菲波那契数列（Fibonacci 序列）的前 20 项。



The screenshot shows a C++ IDE with a project named "E:\Code\Kariga\25C\小作业\4\7斐波那契.c". The code defines a recursive function `Fibonacci` to calculate the 20th term of the Fibonacci sequence. The output window displays the first 20 terms of the sequence, with the 20th term being 6765. The process exited after 5.052 seconds.

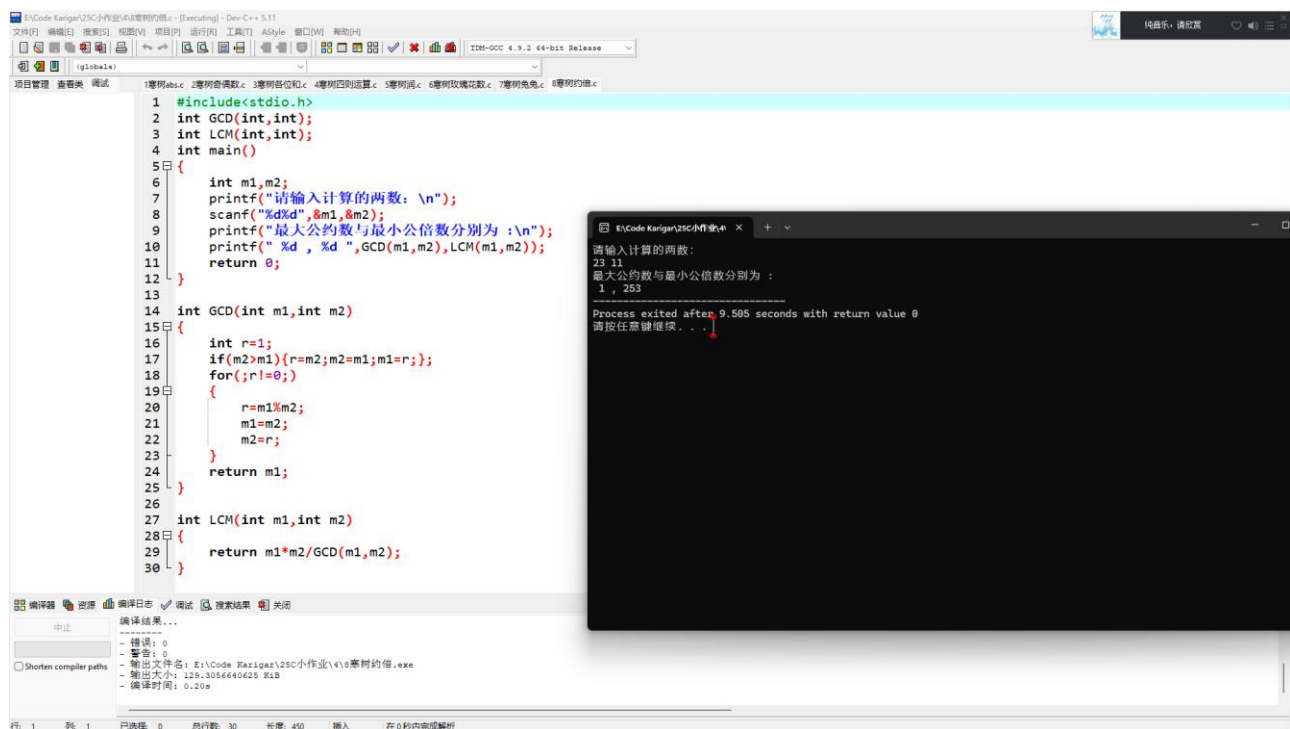
```
1 #include<stdio.h>
2 int Fibonacci(int);
3 int main()
4 {
5     int i,Ent=1;
6     printf("斐波那契前二十项为: \n");
7     for(i=1;i<=20;i++)
8     {
9         printf("%d\t",Fibonacci(i));
10        if(Ent%5==0)
11            printf("\n");
12        Ent++;
13    }
14    return 0;
15 }
16
17 int Fibonacci(int n)
18 {
19     if(n==1||n==2)
20         return 1;
21     else
22         return Fibonacci(n-1)+Fibonacci(n-2);
23 }
```

斐波那契前二十项为:

1	1	2	3	5
8	13	21	34	55
89	144	233	377	610
987	1597	2584	4181	6765

Process exited after 5.052 seconds with return value 0
请按任意键继续...

8、利用函数调用求两整数的最大公约数和最小公倍数。



The screenshot shows a C++ IDE with a project named "E:\Code\Kariga\25C\小作业\4\8求两数的最大公约数和最小公倍数.c". The code defines two functions, `GCD` and `LCM`, to calculate the greatest common divisor and least common multiple of two integers. The output window shows the input values 23 and 11, and the calculated GCD (1) and LCM (253). The process exited after 9.595 seconds.

```
1 #include<stdio.h>
2 int GCD(int,int);
3 int LCM(int,int);
4 int main()
5 {
6     int m1,m2;
7     printf("请输入计算的两数: \n");
8     scanf("%d%d",&m1,&m2);
9     printf("最大公约数与最小公倍数分别为: \n");
10    printf(" %d , %d ", GCD(m1,m2), LCM(m1,m2));
11    return 0;
12 }
13
14 int GCD(int m1,int m2)
15 {
16     int r=1;
17     if(m2>m1){r=m2;m2=m1;m1=r;};
18     for(;r!=0;){
19         r=m1%m2;
20         m1=m2;
21         m2=r;
22     }
23     return m1;
24 }
25
26
27 int LCM(int m1,int m2)
28 {
29     return m1*m2/GCD(m1,m2);
30 }
```

请输入计算的两数:

23 11

最大公约数与最小公倍数分别为 :

1 , 253

Process exited after 9.595 seconds with return value 0
请按任意键继续...

5. 程序及测试结果

6. 实验分析与体会

通过本次实验，我系统地掌握了 C 语言中函数调用的核心应用方法，深入理解了结构化编程的优势与实现路径。在完成八项编程任务的过程中，不仅巩固了基础语法知识，更重要的是学会了如何通过函数模块化地解决复杂问题。

在实验过程中，我深刻体会到函数作为程序基本单元的重要性。通过绝对值函数、四则运算函数等实现，理解了函数参数传递与返回值的机制。特别是在开发四则运算程序时，发现将不同运算封装为独立函数后，主程序的逻辑变得异常清晰，这种模块化设计大大提升了代码的可读性和可维护性。在玫瑰花数判断程序中，通过将指数计算单独定义为 Exponent4 函数，有效避免了代码冗余，这种分层设计思维对提升编程效率有显著帮助。

递归算法的实践让我对程序执行机制有了新的认知。在实现斐波那契数列时，虽然递归写法简洁优雅，但通过调试过程直观感受到递归调用的堆栈展开过程，这加深了对程序内存管理的理解。同时，在最大公约数（GCD）和最小公倍数（LCM）的计算中，将辗转相除法封装为独立函数并与 LCM 函数形成调用关系，这种函数嵌套的设计方式展现了代码复用的精妙之处。

数据处理能力的提升是本次实验的重要收获。在数字各位求和、四位数分解等任务中，通过取余运算和循环结构的配合，掌握了数值处理的基本方法。闰年判断程序让我认识到逻辑运算符在复杂条件判断中的关键作用，而玫瑰花数检测则展示了数学建模与编程实现的完美结合。

通过本次实验，我不仅熟练掌握了函数定义、调用和返回值的应用技巧，更重要的是建立了结构化编程的思维方式。在调试过程中，通过逐步验证各函数模块的正确性，培养了系统化的问题排查能力。这些经验为后续学习更复杂的算法和数据结构奠定了坚实基础，也让我对计算机程序设计的艺术性有了更深层次的体会。

实验日期： 2025 年 4 月 14 日

教师评语

签名：

年 月 日